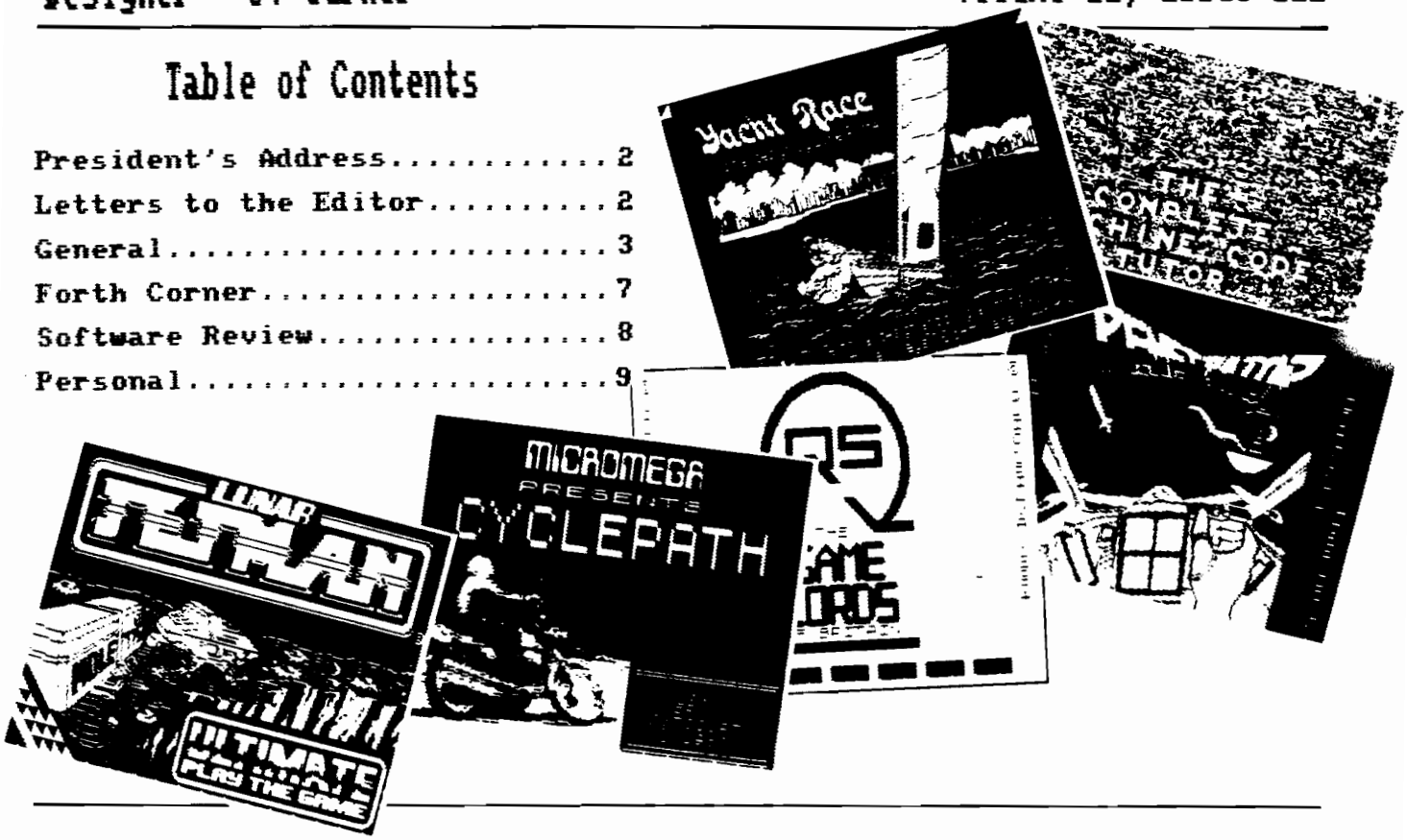
Timex Sinclair User Group NEWSLETTER

Editor - R. Mulder
Designer - J. Turner

Ottawa Chapter
Volume II, Issue III

Table of Contents

President's Address.....2
Letters to the Editor.....2
General.....3
Forth Corner.....7
Software Review.....8
Personal.....9



MEETINGS

LOCATION All meetings are held at the NRC bldg. 100 Sussex Dr. Room 1129B. Check in with security.

TIME Meetings are held at 8:00 p.m. on the last Thursday of each month unless otherwise posted.

FORMAT Each meeting will take the following format:

* President's address and introduction of new members.
* Software Review (Max. 20 min)
* Hardware Review (Max. 20 min)
* Adjournment of formal part of meeting.
* Members informaly discuss common problems, swap ideas etc.....

Advertising Rates

MEMBERS- 45 words or less		FREE
OUTSIDE- 45 words or less		$5.00
COMMERCIAL-		
	full page	$30.00
	half page	$22.00
	quarter page	$12.00
	eighth page	$8.00

President's Address

Here it is the end of May again and summer is just around the corner. I don't know about the rest of you but the demands of running a house at this time of year leaves me with precious little time to spend on my computer. If it isn't painting it's planting or some other chore off the list the warden prepared during the winter. There is something to be said about living in a small apartment.

At the April meeting, several committees were established to improve the operations of the Club. Roger Glover et al have begun the work necessary to provide the membership with a course in BASIC language programming and the target date for the first session is July 1985. The machine language course will commence when the BASIC course is finished. You will recall that the courses will be part of the regular monthly meeting. If you would like to be a "guest lecturer", get in contact with Roger.

Bob Mulder will head up a committee to set the rates and format for advertising in the Club newsletter. Dave Solly will conduct a survey to determine which magazines and newsletters the Club should subscribe to and K. Schiemann will draft a membership-drive flyer and investigate other avenues to encourage other TIMEX/SINCLAIR owners to join the Club.

The membership approved the purchase of a power bar ($25) and additional blank cassettes ($25) for the library.

I was asked by several members to purchase keyboards from Century 21 Electronics. (Note: I was able to obtain 11 keyboards at $10 each and Victor Shideler is now organizing a time and place where the members involved can wire the keyboards to their computers.)

Many thanks to Roger Glover for his lecture on structured programming; to Mike Dove for his hardware demonstration on the Smith-Corona printer/TS2068/TASWORD Interface system, and to Alan Moore for his machine code presentation on ways to manipulate the TS1000 display file. If feasible, presenters should provide Bob Mulder with a paper describing their presentation so that it can be included in the newsletter.

Paid-up membership has now passed the 30 mark and there is now sufficient money available to purchase additional demonstration equipment, and software for the Club library. Both of these topics will be discussed at the next meeting.

On another matter, you will recall that I and four other members placed an order for TS2068 disc systems from AERCO with delivery promised three weeks thereafter. After the three weeks passed and no systems were delivered, a call was placed and the order was promised in two weeks. When the equipment still did not arrive, another call was made and the supplier promised delivery in two weeks. This process went until finally after 10 weeks passed the order was cancelled. I fail to see how any supplier can operate this way and remain in business.

See you at the next meeting.

LETTERS TO THE EDITOR

TO ALL EDITORS.....

For the past number of months a few copies of our newsletter have been sent free of charge to various newsletters throughout Canada and the U.S.A. The original idea had been a Newsletter exchange or failing that , specific input to our newsletter from members or subscribers of other newsletters .

Todate, with the exception of a few more concerned organizations such as Gary Smith and his XFORTH XCHANGE and personal individuals such as John Oliger, we have, UNFORTUNATELY,

received not a scrap of paper, newsletter or explanation (in the case of a number of papers where an agreement had to my knowledge been arrived at).

To say that I am disappointed is to understate the case. I am hoping that there is a satisfactory answer to this problem, as I had felt that co-operation amoungst the few of us that remain would be to everyone's benefit.

I would appreciate hearing from those of you who had or now offer Newsletter exchanges and maybe resolve this misunderstanding to everyone's mutual satisfaction.

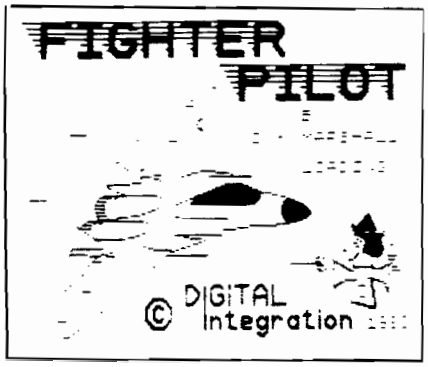
As many of you can see, our Newsletter is evolving into a going concern. However this cannot continue unless it is supported by as many as possible. We have by virtue of this copy of the Newsletter as well as those that were sent previously, supported your interests also.

Please help us to help you! Please write even if it is to notify us that the agreements are off.

Thank-you for your future support!!!

The Editor,
TSUG NEWSLETTER
Timex/Sinclair
Users Group
Ottawa/Hull Ch.
c/o 543 Joanisse
Gatineau, Quebec
J8R1H1 Canada

Attn. R.Mulder



GENERAL

TRANSFERRING MACHINE CODE FROM A TS1000/ZX81 REM STATEMENT TO THE TS2068 USING UPLOAD 2000 OR FIRSTLOADER

You've probably noticed that if a TS1000/ZX81 contains machine code it will not be converted if you use UPLOAD 2000 or FIRSTLOADER. This article will permit you to save the code if it is located in a TS1000/ZX81 REM statement.

1. The first step is to load UPLOAD 2000 or FIRSTLOADER. To stop the program from converting TS1000/ZX81 tokens to TS2068 tokens, POKE 45013,201. Load the TS1000/ZX81 program using RAND USR 45000 as usual. When the program stops, type PRINT (PEEK 49279+256*PEEK 49280) to determine the number of bytes of code in the REM statement. If you know this ahead of time, all the better. Lets say the number of bytes is 1200. Assign a name for the code statement, say, TGIF, then in COMMAND mode type SAVE "TGIF" CODE 49284,1200. Put a blank tape in your dataset and start recording. Press the ENTER key. Your machine code has now been saved.

2. Needless to say that this machine code will have to be changed to match the new address where it has been saved, or any other location where you may wish to store it. Similarly all calls in the code will have to amended to accomodate the new addresses. If you're lucky, relative addressing was used in the code and no calls were made to the TS1000/ZX81 ROM.

3. If your program also contained BASIC statements, make a copy of the tape on your TS1000/ZX81 with the REM statement containing the machine code deleted and do the conversion of the BASIC part using UPLOAD 2000/FIRSTOADER in the normal way. The BASIC and machine code parts of the program can then be combined on a single tape. NOTE: Any calls to the machine code from BASIC will have to be changed to match the new machine code addresses.

4. As an alternative method of deleting the code in the REM statement, POKE zeroes in all addresses starting at 49284 to

49284 + the number of bytes of code in the REM statement less one. Use this method following Step 1. After zeroes have been entered, type POKE 45013,205 then RAND USR 45013 to complete the conversion. You can use a BASIC program to POKE zeroes such as:

```
10 FOR x=49284 TO ?????
20 POKE x,0
30 NEXT x
```

Delete this program before you type RAND USR 45013 as indicated above.

I have not tried this method but I see no reason why it wouldn't work.

Timex Sinclair Computers and Amateur Radio

-by Henry McGee (WE3ACF)

Many of us have acquired our computer for various reasons such as games,music, programs and word processing.Computers have also revolutionized radio communications by replacing the old teletype machines and even the time honoured art of morse code reception.

In the area of amateur radio there has been a steady stream of software to design antennas predict propagation conditions to any part of the world,transmit and receive morse code,teletype,and even to teach code. I have programs which I will make available to the club library which I have obtained from the British Sinclair Amateur Radio Group (SARUG).Two programs now available for TS 2068 or Spectrum are WOTSON for radio propagation and TUTOR as a morse code learning aid.

The following program by A.Jones will give computer to computer communication over the air using the TS 2068 or Spectrum.The equivalent ZX 81 or TS 1000 program is also shown.The program is called Sinclair High Speed Teletype (SHTTY) or, in the case of the ZX 81,ZXTTY.All connections are through the mic and ear jacks.

```
 TS 2068
2 PRINT D$:LET D$=" "
4 PAUSE 0 :LET A$=INKEY$ :
  IF A$= " STOP " THEN GOTO 8
5 BEEP .02,34   LET D$=D$ + A$
  PRINT A$;
6 GOTO 6-2*(INKEY$ = " ")
8 SAVE "" LINE 1 : LOAD "" :
  (note that STOP is symbol
shift A)
Enter LET D$ = " "   GOTO 3.
```

Connect a cassette and save as normal,rewind & the program reloads itself.Anything typed it is printed onscreen, using symbol shift 5 as backspace and ENTER as newline.Pressing STOP saves the program.

Now, connect the ear jack to the receiver and the mic jack to the transmitter microphone input (non amateurs can duplicate this with a cassette recorder).Have the other party connect his computer in the same way and type LOAD "" : ".Type in your own message ,press PTT for turn on the cassette recorder and hit STOP. The message will appear on the screen at the other end and the other party will have the program ,just like a return envelope.He can now reply using the same process.

This system has been tried on 2 metres FM and any signal over about S7 may be received well. Keep the AF-gain quite high.

The same program for the ZX 81 is as follows

```
10 LET M$="**"
50 CLS
60 FOR I=1 TO LEN M$
70 IF M$(I)="**" THEN PRINT"**
80 IF M$(I) <> "**" THEN PRINT
   M$(I);
90 NEXT I
100 PRINT TAB,"----------------
-----------------TIME.CALL...
MESSAGE"
120 INPUT Z$
130 PRINT Z$;  TAB    0;"(S=SEND
C = CHANGE Q= QUIT)"
150 INPUT W$
160 IF W$ = "Q" THEN STOP
190 IF W$ <> "S" THEN GOTO 50
200 LET M$ = M$ + Z$ + "**"
210 SAVE "Q50"
220 GOTO 50
```

I hope to have some more interesting amateur radio applications for the next issue. In the meantime, if you want any more information on the above,just give me a call at 521 8640 in the evening. 73.

Source of Sinclair ROM to change your TS 2068 to a Spectrum

COMBINED PRECISION COMPONENTS
DEPT . P,190-200 NORTH ROAD
PRESTON, LANCASHIRE
PR1 1YP , ENGLAND

The cost for the ROM is £4.69 plus £3.00 for postage and handling, or about $13.00 (C).

- Henry McGee

ZX81 UTILITY GENERAL ROUTINES

Rounding a number to one decimal place
LET X= (INT(10*(X+0.05))/10
Rounding a number to two decimal places
LET X= (INT(100*(X+0.005))/100
Rounding to one decimal place if this will not make a fraction vanish
IF INT X>0 THEN LET X= (INT(10*(X+0.005))/10
To find the common or base 10 logarith of a number X (only natural or base e logarithm is built into the computer)
LET L= (LN X)/(LN 10)
Getting the constant e
LET E= 2.7182818
The Natural anti-logarithm of X
LET A= 2.718218 ** X
The ZX81 will accept scientific notation in the following form:
3.789 * 10↑7 is written
3.789E7
Or instead of * 1000000 one can write * 1E6

TIP:

Frequently you want to save your program two or three times. This usually means sitting at your computer's side while it saves itself once, then getting into the program's autosave routine again, etc. The listing that follows will prompt you for the number of saves you want, and then will save the program the specified number of times.

The beauty of the routine is that each save will auto-run when loaded. The routine is based on the observation that the ZX-81 always resets certain memory locations to zero when it "NEWS" itself, or when it loads a new program. The routine stores the number of saves in this location, and then checks to see if it is zero. If not, then the position is decremented and another save occurs. If it is zero the program runs itself. Since the location will always be zero when the program is loaded, each save will always auto-run.

For a 64K program, I use memory location 65000. For a 16K program try location 16000.

```
9900 CLS
9905 PRINT AT 10,0; "HOW MANY SAVES ?"
9910 INPUT X
9915 PRINT AT 15,0; "START RECORDER,
     PRESS ANY KEY WHEN READY."
9920 IF INKEY ="" THEN GOTO 9920
9930 CLS
9935 POKE 65000,X
9940 POKE 65000,(PEEK 65000)-1
9945 SAVE "PROGRAM"
9950 IF NOT PEEK 65000 THEN GOTO 1
9955 GOTO 9940
```

- (ON PAGE

ADVERTISING RATES:

(effective - May/85)

Members of the TSUG OTTAWA/HULL Chapter have exclusive right to run personal ads related to TSUG free of charge for up to but not exceeding 45 words.

OUTSIDE personals will be accepted at a flat rate of $5.00 for up to but not exceeding 45 words. Ads for personals greater than 45 words will be charged commercial rates.

COMMERCIAL RATES (per issue)

1	full page	$20
1/2	page	$22
1/4	page	$12
1/8	page	$8

FORMAT REQUIRED

Personals- membership either on cassette using: Tasword (32 char.) Word II (ZX81) or: on 2040 paper or 8x11 bond formatted and checked

Outside- Typed on 8x11 standard bond formatted and checked

Commercial (copy ready)
- for full and 1/2 page ads as company requires,no restriction
- for 1/4 or 1/8 ads, formatted in 32 char columns same as outside ads.

Payment

Must accompany ads submitted Check or money order payable to

TSUG OTTAWA/HULL
c/o The Editor
648 Joanisse ST.,
Gatineau,Quebec
J8R 1H1 Canada

ME AND MY PRINTER

Several months ago I decided that I would like to use my 2068 as a wordprocessor. I was familiar with the two word processors M-SCRIPT and TASWORD. I felt that TASWORD would be better for my purpose particularily since it was more reliable than M-SCRIPT in its loading and saving mode. Inially I decided to deal with BYTE-BACK and to purchase their interface for their Banana-Gorilla printer. It did not take more than a month for the interface and driver software to arrive. However it took close to three months for the printer to arrive. Apparently they did not have any in stock and it had to be purchased from another source. However when I had everything in my hands I found out to my dismay that I could not operate the driver program with the software TASWORD. I phoned BYTE-BACK and learnt that they too did not know how to make the two of them compatable. Apparently they were working on the problem. Recently (the third week in April) I was contacted by BYTE-BACK and told that the problem was solved and that I would shortly be receiving the solution to the problem. Needless to say I was quite frustrated. So I phoned Gladstone Electronics in Toronto and ordered from them a Tasman interface and a Smith-Corona D100 printer. Within the week I had everything in my hands and surprisingly it worked flawlessly. After the fact I learnt that I could have purchased the identical printer here in Ottawa for about $100 cheaper.

I was charged $169.95 for the Tasman interface (this includes the interface, cable, and driver software), $499.95 for the Smith-Corona D100 printer, sales tax and shipping charges. As I already mentionned the printer could be purchased here in Ottawa for about $100 less and I am sure that if I were not so pressed for time that I could have found a cheaper source for the interface.

TASWORD contains the driver software to run the printer and it works flawlessly. The driver program which comes with the interface allows you to LPRINT, LLIST or COPY. Printed below is an example of the various fonts.

ENLARGED PRINT

CONDENSED PRINT
UNDERLINED PRINT
ITALICS
EMPHASIZED PRINT
DOUBLE STRIKE
ELITE PRINT
PROPORTIONAL PRINT

I am more than pleased with its performance.

Michael J Dove

DISC DRIVE INTERFACE
T/S 1000 - 2068
Uses Shugart 455 compatables
10K in 3 secs.
DOS on EPROM
$115
LARKEN ELECTRINICS
Ottawa, Ontario
728-5924

RAMEX INTERNATIONAL LTD
Utica, Michigan
U.S.A.
* Tasword II
* Millinium K Disc Drive System
See Library for copy of catalogue.

the JOHN OLIGER COMPANY
11601 Whidbery Dr.
Cumberland, In. 46229
TS1000/2068 HARDWARE PRODUCTS
see full ad at back of newsletter

CURRY COMPUTER
5344 WEST BANFF
GLENDALE AZ.
U.S.A.
T/S1000 software
T/S2068 software and hardware
see club library for copy of catalogue.

THE ABOVE IS A SAMPLE OF POSSIBLE ADS ONLY USING REAL COMPANIES. (I'M SURE THEY WILL NOT MIND TOO MUCH)

4TH CORNER

Well here it is again and no input from those dedicated forth users who use the language but don't like to share their experiences. I guess that I will have to be the one man snow again. The only reason that I would continue this 'act' is the fact that I am a truely dedicated Forth hacker (or dumb idiot who doesn't know when to call it quits).

What I am going to do is, in essence, to develop a small game in Forth, patterned after a Basic game called MASTERMIND. The game is such that the computer chooses 4 colours out of 8 in a particular order and the player must guess what the colours are in the proper order.

The method I am going to take will be the development of a core and expand, change or improve on this core upto the point of a complete compact game program.

There will be obvious areas where improvements, corrections, additions, or even a few of the next steps to be taken will become obvious. This is the place where you the supposed Forth user comes in. I hope to receive appropriate input to the core or questions or anything that would suggest that I am not doing this for nothing. In other words.....HELP!!!!

Each installment of the program will take approximately 2 screens of pure program data. Comments to the program to help identify the functions will be printed as a seperate block.

FORTH MIND
A FORTH STRATEGY GAME
EXPLANATIONS-FORTH MIND

```
ARRAY- set up an array defining
       word
PLAYER- set up an array to
        capture the player's
        input
COMPUTER- set up an array to get
          the computer's input
COMPARE- set up an array to
         capture a comparison
         of the player's vs
         the computer's choices
SEED- seed value for a rnd #
      generator
SEEDON- set up initial value
        of seed
RAND- establish how the rnd
      function will start
RND- final word for generator
     requires one value on stack
INPUT.COM- computer's 4 colour
           choices
INPUT.COMPARE- compare player's
       choices to the computer's
INDIV.MATCH- variable to capture
       the number of correct
        choices in the correct
        positions
ANY.MATCH- flag to indicate any
         match in any position
DELAY- as word indicates
TRIES- variable to capture the
       number of tries the
       player uses
ECHECK- check player input to
       ensure proper values
INPUT.PLAY- calls for player's
       input
```

End of first screen

```
SCR # 1
  0 ( SCR#1 OF FORTHMIND CORE)
  1 : ARRAY <BUILDS 1+ 2 * ALLOT
 DOES> SWAP 2 * + ;
  2 4 ARRAY PLAYER 4 ARRAY COMP.
TER 4 ARRAY COMPARE 0 VARIABLE S
EED
  3 : SEEDON SEED @ 75 U* 75 0
D+ OVER OVER U< - - 1 - DUP SEED
! ;
  4 : RAND DUP 0= IF 23672 C@ SW
AP DROP THEN SEED ! ;
  5 : RND SEEDON U* SWAP DROP
  6 : INPUT.COM 5 1 DO I 8 RND S
WAP COMPUTER ! LOOP ;
  7 : INPUT.COMPARE 5 1 DO I 2DU
P PLAYER @ SWAP COMPUTER @ = IF
  8 1 I COMPARE ! ENDIF LOOP ;
  9 0 VARIABLE INDIV.MATCH 0 VAR
IABLE ANY.MATCH
 10 : DELAY 2500 0 DO NOOP LOOP
; 0 VARIABLE TRIES
 11 : ECHECK BEGIN KEY 48 - DUP
7 > DUP IF 21 2 AT
 12 ." INCORRECT INPUT! TRY AGAI
N" DELAY SWAP DROP 21 2 AT
 13 27 SPACES ENDIF 0= UNTIL ;
 14 : INPUT.PLAY 5 0 DO ECHECK I
 PLAYER ! LOOP 1 TRIES +! ;
 15 -->
```

Software REVIEW

System: Spectrum 48k
Cost: 7.95 (pounds sterling)
Purpose: Extended Basic Utility

Title:

YS MEGA BASIC

Have you ever run accross a statement or an advertisement that seems at the very least to be an obvious case of overstatement of the facts? Well I thought the same thing when I first read the ad for YS Mega Basic from Your Spectrum magazine. Who could believe MacIntoch windows,sprites,icons, multiple fonts, etc etc on a SPECTRUM no less. Well I being the type to be more than impressed by anyone that would make such an apparent overstatement took the plunge and ordered myself a copy.

Well I am more than pleased to say that I was not disappointed with the utility. Other than a program bug in the initial version which was rectified by a fix published in the following issue of Your Spectrum, the package functions as advertized.

To highlight a few of the added features of YS Mega Basic, the 35 odd new commands encompase such things as:

up to 10 windows, simple multi-tasking, 3 fonts 4 modes of printing, 64 character columns, enlarged characters, interrupt sound generator,amoungst many others.....

In essence YS MEGA BASIC transforms your 2068 cum Spectrum into something resembling an Apple or a Commodore or ewen a MacIntoch. Gone are the token words but in their place is an auto-line numbering system, trace function, line delete and much more. Procedures and Repeat-Until constructs make this a very powerfull basic indeed.

The graphic capacilities are greatly improved and make this package a dream, scroll, pan, sprite operations, inversions and other functions enhance the presentation of any program using this program.

You can examine, alter run,or do many other things with the 'frontpanel' in essence a full spec disassembler.

A spite designer is included as a seperate program written in Mega Basic and displayes the obvious advantage of the languages versatility.

What more can I say on the good side--what more could you want??

YS MEGA BASIC is fully supported via the magazine Your Spectrum and programs and techniques will be published on an ongoing basis.

The only real problem with this package is that if you write a program using Mega Basic, to use it again in the future requires that you must load 2 programs each time you need it. The other item is that who will use it if others do not have Mega Basic. Your Spectrum has indicated support of a kind via publication in their mag. Is it possible to have a compiled version of your program with Mega Basic thereby loading only one program? Who knows!

In any event, in my personal opinion, I beleive this utility to be the ultimate and if your Spectrum (2068 with Romswitch) is not Mega Basic driven then what good is it?

I call them like I see them!

R.Mulder

HARDWARE REVIEW

HARDWARE PROJECT

See hardware project at end of newsletter.....
SOUND GENERATOR FOR THE ZX81

R.GLOVER

PERSONAL

HOME INVENTORY

ZX81

```
   5 REM HOME INVENTORY
  10 REM THIS PROGRAM WILL ALLOW
 YOU TO ITEMIZE YOUR INVENTORY
OF HOUSEHOLD ITEMS IN THE EVENT
OF FIRE OR THEFT.
  15 DIM A$(150,18)
  20 DIM B$(150,18)
  25 DIM C$(150,18)
  30 DIM D$(150,9)
  35 DIM N$(1,9)
  40 LET A=0
  50 LET R$="HOME INVENTORY"
  55 GO SUB 2000
  60 PRINT AT 5,5;"SELECT:";AT 7
,6;"1) MAKE AN ENTRY?";AT 9,6;"2
) SAVE YOUR LIST?";AT 11,6;"3) C
HECK YOUR LIST?";AT 13,6;"4) LIS
T BY LOCATION?";AT 15,6;"5) LIST
 BY ARTICLE?";AT 17,6;"6) LIST T
O PRINTER?"
  65 GO SUB 3500
  70 GO SUB 100*R
  75 GO TO 50
 100 REM MAKE AN ENTRY
 105 FOR X=A+1 TO 150
 110 LET A=A+1
 115 LET R$="MAKE AN ENTRY"
 120 GO SUB 2000
 125 FOR Y=1 TO 7
 130 GO SUB 2380+20*Y
 135 NEXT Y
 140 LET R$="MAKE AN ENTRY"
 141 GO SUB 2000
 142 GO SUB 3200
 143 GO SUB 3600
 144 IF R$="Y" THEN GO TO 152
 146 LET B=1
 148 GO SUB 352
 150 LET B=0
 152 LET R$="MAKE AN ENTRY"
 154 GO SUB 2000
 156 PRINT AT 10,3;"ANYMORE ENTR
IES? (Y OR N)"
 158 INPUT R$
 160 IF R$="N" THEN GO TO 50
 162 NEXT X
 163 CLS
 164 PRINT AT 10,2;"SORRY, THE C
OMPUTER IS FULL"
 166 PAUSE 200
 168 GO TO 50
 200 REM SAVE LIST
 210 LET R$="SAVE LIST"
 220 GO SUB 2000
 230 PRINT AT 10,0;"REWIND TAPE,
 START RECORDING.    WAIT 10 SECO
NDS THEN PRESS       ENTER  KEY"
 240 INPUT R$
 250 CLS
 260 SAVE "INVENTORY"
 270 GO TO 50
 300 REM CHECK ENTRIES
 304 FOR X=1 TO A
 308 LET B=0
 312 LET R$="CHECK ALL ENTRIES"
 316 GO SUB 2000
 320 GO SUB 3200
 324 GO SUB 3600
 332 IF R$="N" THEN GO TO 352
 336 CLS
 340 IF B=1 THEN GO TO 50
 344 NEXT X
 348 GO TO 50
 352 GO SUB 3615
 354 LET B=1
 356 PRINT AT 12,0;,"WHAT INFOR
MATION IS WRONG?",,"1=LOCATION"
,"2=ARTICLE","3=MODEL NO.",,"4=SE
RIAL NO.",,"5=MANUFACTURER","6=PU
RCHASE DATE","7=COST","8=NOTHING
-EXIT","9=DELETE ENTRY"
 360 INPUT Y
 364 IF Y=8 THEN GO TO 50
 366 IF Y=9 THEN GO TO 2900
 368 GO SUB 2380+20*Y
 372 GO TO 320
 400 REM LIST BY LOCATION
 404 LET R$="LIST BY LOCATION"
 406 GO SUB 2000
 408 LET R$="LOCATION"
 410 LET Q=0
 412 GO SUB 2100
 425 LET N$(1)=Z$
 436 FOR X=1 TO A
 440 IF N$(1)=A$(X, TO 9) THEN G
O SUB 462
 444 NEXT X
 448 IF Q=0 THEN GO TO 480
 456 GO TO 50
 462 GO SUB 2000
 464 GO SUB 3200
 466 GO SUB 3000
 468 LET Q=1
 472 RETURN
 476 GO TO 50
 480 PRINT AT 10,7;"ENTRY NOT FO
UND"
 482 GO SUB 3000
 484 GO TO 50
 500 REM LIST BY ARTICLE
 505 LET R$="LIST BY ARTICLE"
 510 GO SUB 2000
 511 LET R$="NAME OF ARTICLE"
 512 GO SUB 2100
 514 LET Q=0
 515 LET N$(1)=Z$
 520 FOR X=1 TO A
 525 IF N$(1)=A$(X,10 TO ) THEN
GO SUB 460
 530 NEXT X
 535 IF Q=0 THEN GO TO 480
 540 GO TO 50
 600 REM LIST TO PRINTER
 605 LET R$="LIST TO PRINTER"
 610 FOR X=1 TO A
 615 IF A=0 THEN RETURN
 630 LPRINT "LOCATION      ";A$(
X, TO 9),"ARTICLE      ";A$(X,1
0 TO ),"MODEL NO.     ";B$(X, T
O 9),"SERIAL NO.    ";B$(X,10 TO
 ),"MANUFACTURER  ";C$(X, TO 9)
,"PURCHASE DATE:";C$(X,10 TO ),
"COST          ";D$(X, TO 9)
 640 NEXT X
 650 RETURN
2000 CLS
2010 PRINT "█▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄█"
2020 PRINT "█
█"
2030 PRINT "█▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀█"
2040 PRINT AT 1,15-.5*LEN R$;R$
```

```
2050 RETURN
2100 PRINT AT 21,0;"                                  "
2105 LET R$="ENTER "+R$
2110 PRINT AT 21,15-.5*LEN R$;R$
2320 INPUT Z$
2340 IF LEN Z$<=9 THEN RETURN
2360 PRINT AT 21,0;"      TOO MAN
Y CHARACTERS                  "
2370 PAUSE 150
2380 PRINT AT 21,0;"                                  "
2390 GO TO 2110
2400 LET R$="LOCATION"
2405 GO SUB 2100
2410 LET A$(X, TO 9)=Z$
2415 RETURN
2420 LET R$="NAME OF ARTICLE"
2425 GO SUB 2100
2430 LET A$(X,10 TO )=Z$
2435 RETURN
2440 LET R$="MODEL NUMBER"
2445 GO SUB 2100
2450 LET B$(X, TO 9)=Z$
2455 RETURN
2460 LET R$="SERIAL NUMBER"
2465 GO SUB 2100
2470 LET B$(X,10 TO )=Z$
2475 RETURN
2480 LET R$="MANUFACTURER"
2485 GO SUB 2100
2490 LET C$(X, TO 9)=Z$
2495 RETURN
2500 LET R$="PURCHASE DATE (25/0
9/83)"
2505 GO SUB 2100
2510 LET C$(X,10 TO )=Z$
2515 RETURN
2520 LET R$="COST"
2525 GO SUB 2100
2530 LET D$(X)=Z$
2535 RETURN
2900 LET A$(X)=A$(A)
2905 LET B$(X)=B$(A)
2910 LET C$(X)=C$(A)
2912 LET D$(X)=D$(A)
2915 LET A$(A)="  "
2920 LET B$(A)="  "
2925 LET C$(A)="  "
2927 LET D$(A)="  "
2930 LET A=A-1
2935 GO TO 312
3000 PRINT AT 21,3;"PRESS  ENTER
 TO CONTINUE"
3010 INPUT Z$
3020 CLS
3030 RETURN
3200 PRINT AT 4,0;
3210 PRINT "LOCATION        ";A$(X
, TO 9);"ARTICLE        ";A$(X,10
 TO );"MODEL NO.        ";B$(X, TO
 9);"SERIAL NO.        ";B$(X,10 TO
);"MANUFACTURER ";C$(X, TO 9);
"PURCHASE DATE";C$(X,10 TO )
COST          ";D$(X, TO 9);
3230 RETURN
3400 INPUT R$
3405 LET R=0
3410 IF LEN R$<=9 THEN RETURN
3415 LET R=1
3420 PRINT AT 21,0;"      TOO MA
NY CHARACTERS                 "
3425 PAUSE 150
3430 PRINT AT 21,0;"                                  "
3435 GO TO 3400
3500 INPUT R
3505 IF R<=8 THEN RETURN
3510 PRINT AT 21,0;"                  INCO
RRECT ENTRY                   "
3520 PAUSE 150
3530 PRINT AT 21,0;"                                  "
3540 GO TO 3500
3600 PRINT AT 21,0;"      ENTER C
ORRECT (Y OR N)?              "
3610 INPUT R$
3615 PRINT AT 21,0;"                                  "
3620 RETURN
```

REM

ZX-81 Utility
REM Extender

Purpose: Extend REM line and will not disturb current contents

HEX LISTING:

```
E7 E7 CD D9 14 CD 8A 15
60 69 CD D8 09 28 03 CF
72 00 23 23 E5 4E 23 46
C5 09 E5 E7 CD D9 14 CD
8A 15 E1 C5 CD 9E 09 C1
E1 09 EB E1 73 23 72 EB
ED 42 19 EB 21 0C 00 37
3E 1B E5 EC 42 E1 D0 EB
77 23 0B EB 18 F4
```

TOTAL BYTES=70
TO USE: LOAD TO ANY CONVENIENT ADDRESS
TO CALL:

 :If Usr (your address) Then REM (line num);bytes to add:

eg.:If Usr 9668 Then REM 12,250 will add 250 bytes to the end of line number 12.

Report codes Inverse 7 = no such line exists
4 = not enough memory
0 = success/ok

- F. Curzon

EARTH-ATTACK

48K

```
   1 CLEAR 37299
   2 BEEP .5,12: BEEP .5,24
  30 GO SUB 1000
  40 GO SUB 2000
  50 BORDER 0
  55 LET s$="...."
  60 LET hs=0
  65 LET s=0
  70 LET t=0
  80 LET bo=0
  85 GO TO 3000
  99 PAPER 0: INK 7: CLS : FOR n
=0 TO 20: BEEP .005,n: NEXT n
 100 LET x=9: LET y=5: LET b=1
 110 FOR n=0 TO 7: PRINT INK 1 A
T n,0;"[redacted]": NEXT n
 120 LET s=0
 130 PRINT AT 20,0;"SCORE:",s
 140 LET a$="A A A A ": PRINT AT
 20,15;a$
 150 FOR n=16 TO 19: PRINT AT n
0; INK 5;"[redacted]": NEXT n
 160 LET l=20
 165 PRINT AT 0,0;"LASERS:";l;"
"
 170 PRINT AT 0,16;"HI-SCORE:";h
s
 180 PRINT AT 18,15; INK 2; PAPE
R 5;"DDD": LET sb=3
 200 LET x=x+(IN 64510<31)-(IN 6
3486<31): PRINT AT x,y;"A";
 210 LET p=PEEK (PEEK 23684+256+
PEEK 23685): IF p THEN GO SUB 50
0
 220 IF IN 61438<31 THEN IF l TH
EN GO SUB 400
 230 IF RND>.79 THEN PRINT AT RN
D+5+6,31: INK 5;"B"
 240 IF IN 32766<31 THEN GO SUB
700
 245 LET rg=RND: IF rg>.96 THEN
PRINT AT 15-b,31: INK 4,("C" AND
 rg>.98)+(CHR$ 16+CHR$ 6+"F" AND
 rg<=.98): BEEP .005,10: BEEP .0
05,20
 255 LET b=INT (RND*3)+1
 260 PRINT AT x,y;" ": POKE 3747
7,b: LET a=USR 37426: IF IN 6527
8<31 THEN LET a=USR 37426
 270 GO TO 200
 400 REM Firing Subroutine
 405 LET l=l-1
 440 BEEP .005,30: LET a=USR 375
07: IF a THEN BEEP .01,7: BEEP .
01,0: GO TO 450
 442 PRINT AT 0,7;l;" "
 445 RETURN
 450 LET s=s+10*a
 460 PRINT AT 20,6;s;" ";AT 0,7;
l;" "
 470 RETURN
 500 IF p=60 THEN : LET l=l+20
BEEP .01,20: GO TO 450
 501 BEEP .01,1: BEEP .01,2: BEE
P .01,4: BEEP .01,8: BEEP .01,16
: BEEP .01,8: BEEP .01,4: BEEP .
01,2: BEEP .01,1
 510 PRINT AT x,y; OVER 1;"#": B
EEP .01,0: BEEP .01,-.5: BEEP .0
1,-1
 515 IF a$="!" THEN GO TO 600
 520 LET a$=a$(3 TO ): IF a$="
THEN LET a$="!"
 530 LET s=s-10: PRINT AT 20,15;
a$;"  "
 550 PRINT AT x,y; INK 1;"[redacted]": LE
T a=USR 37426: LET x=9
 560 BEEP .3,0: PRINT AT 20,5;s;
" ": RETURN
 600 PRINT AT 4,10; FLASH 1;"GAM
E OVER": FLASH 0: FOR n=1 TO 3:
LET a=USR 37300: BEEP .1,n: NEXT
 n: PAUSE 2: PAUSE 1: IF hs<s TH
EN LET hs=s: GO SUB 800
 610 GO TO 3000
 620 GO TO 100
 700 LET sb=sb-1
 701 IF INT (s/1000)>bo THEN LET
 sb=3: PRINT AT 18,15; INK 2; PA
PER 5;"DDD"
 702 IF sb<0 THEN RETURN
 710 LET s=s+10*USR 37400: BEEP
.5,7: BEEP .5,0: PRINT AT 20,6;s
;" ": PRINT AT 18,15+sb; INK 5;"
[redacted]"
 715 IF INT (s/1000)>bo THEN LET
 sb=3: PRINT AT 18,15; INK 2: PA
PER 5;"DDD"
 716 LET bo=INT (s/1000)
 720 FOR n=0 TO 31: POKE 37477,I
NT (RND*3)+1: LET a=USR 37426
 725 PRINT AT x,y-1; PAPER 8;" A
": BEEP .01,n
 730 NEXT n: RETURN
 800 PAPER 0: INK 6: CLS
 810 PRINT "          EARTH ATTAC
K"
 820 LET a$="WELL DONE, YOU HAVE
 BEATEN THE  HIGH SCORE; PLEASE
ENTER YOUR  INITIALS NOW"
 830 PRINT ''
 840 FOR n=1 TO LEN a$: PRINT a$
(n);: BEEP .003,0: NEXT n
 850 PRINT ''TAB 5;
 860 LET s$=""
 862 PRINT "---";: FOR n=1 TO 3
 PRINT CHR$ 8;: NEXT n
 865 FOR n=1 TO 3
 866 PRINT "?";
 870 PAUSE 0: LET p$=INKEY$: IF
LEN p$<>1 THEN GO TO 870
 880 PRINT CHR$ 8;p$;: LET s$=s$
+p$
 885 BEEP .1,n
 890 NEXT n
 900 PAUSE 50: RETURN
 999 STOP
1000 FOR n=1 TO 6: READ a$
1010 FOR p=0 TO 7: READ a
1020 POKE USR a$+p,a: NEXT p
1030 NEXT n
1040 DATA "a",BIN 11100000,BIN 1
110000,BIN 11111000,127,127,BIN
11111000,BIN 110000,BIN 11100000
1050 DATA "b",BIN 11000,BIN 1001
00,126,129,129,126,0,0
```

```
1055 DATA "c",BIN 111100,BIN 100
0010,129,BIN 1000010,60,60,60,60
1056 DATA "d",0,64,BIN 1011100,1
26,126,BIN 1011100,64,0
1057 DATA "e",255,BIN 10011001,B
IN 10011001,255,BIN 10011001,BIN
 10011001,255,255
1058 DATA "f",BIN 1011101,BIN 10
1010,BIN 110110,127,BIN 1010101,
BIN 1010101,BIN 10100,BIN 110110
1
1060 RETURN
2000 REM SMART BOMB
2010
2020 DATA 33,255,88,14,0,6,255,3
5,126,254
2030 DATA 6,32,7,12,62,8,119,16,
-12,201
2040 DATA 198,16,119,16,-18,201
2050
2060 REM SIDE SCROLL
2070
2080 DATA 6,64,33,0,72,17,32,0,5
2,0
2090 DATA 119,25,16,252,33,0,89,
6,8,17
2100 DATA 32,0,62,7,119,25,16,-4
,1,0
2110 DATA 8,17,0,72,33,1,72,237,
176,1
2120 DATA 0,1,17,0,89,33,1,89,23
7,176
2130
2140 REM BUILD CITY
2150
2160 DATA 6,0,14,15,62,22,215,12
1,215,62
2170 DATA 31,215,62,16,215,62,5,
215,62,17
2180 DATA 215,62,0,215,62,143,21
5,13,16,-26
2190 DATA 201
2200
2210 REM CHECK FIRING
2220
2230 DATA 14,0,42,132,92,17,0,3,
25,229,6,5,126,254,129,40,16,254
,127,40,11,54,255,35,16,-14,118,
118,118,118,24,47,12,197,237,82
2240 DATA 229,205,230,145
2250 DATA 0,66,36,24,24,36,66,0,
118,118,225
2252 DATA 229,205,230,145,153,90
,16,199,227,8,90,153,118,118,225
2260 DATA 205,230,145,0,0,0,0,0,
0,0
2265 DATA 0,193,12
2266 DATA 225,62,5,54,0,35,4,184
,32,-7,6,0,201
2270
2280 REM INITIALIZATION
2300 FOR n=1 TO 199
2310 READ a: POKE 37399+n,a
2320 NEXT n
2330 DATA 62,7,33,0,89,6,255,119
,35,16,-4,118,61,32,-13,6,1,33,2
55,79,14,32,175,126,203,63,119,4
3,13,32,-8,62,71,188,32,-16,16,-
21,201
2340 FOR n=1 TO 39
2350 READ a: POKE 37299+n,a
2360 NEXT n
2370 DATA 6,8,209,26,19,213,119,
17,0,1,25,16,-11,201
2380 FOR n=1 TO 14
2390 READ a: POKE 37349+n,a: NEX
T n
2400 RETURN
3000 INK 7: PAPER 0: CLS
3110 FOR n=0 TO 7: INK n: PRINT
AT 3,9; INK n;"EARTH ATTACK": BE
EP .1,n
3120 PRINT '"Use Keys " TAB 9;"1
-5......Up" 'TAB 9;"Q-T......Dow
n" 'TAB 9;"6-0.......Fire"''TAB 9
;"SHIFT....Extra Speed"''TAB 9;"
SPACE....Smart Bombs"
3121 PRINT ''"PRESS 'S' TO START"
3122 NEXT n
3125 PAUSE 250
3126 IF INKEY$="s" THEN GO TO 99
3127 CLS
3130 PRINT "Earth Attack"': PRINT
'''
3140 LET a$="B.........10 POINTS"
+CHR$ 13+CHR$ 13+"F.........20 PO
INTS"+CHR$ 13+CHR$ 13+"C.........
REFUELLING Stn"+CHR$ 13+CHR$ 13+
CHR$ 13+"PRESS 'S' TO START"
3150 FOR n=1 TO LEN a$
3160 PRINT a$(n);: BEEP .003,0:
NEXT n
3170 PAUSE 250: IF INKEY$="s" TH
EN GO TO 99
3180 CLS
3190 PRINT "EARTH ATTACK"
3200 PRINT "------------"
3205 PAUSE 50
3210 PRINT '''"HIGH SCORE = ";s:
 BEEP .1,9: PAUSE 50
3230 PRINT ''"HIGH SCORE BY ";s$
: BEEP .1,8
3234 PRINT '''''"PRESS 'S' TO ST
ART"
3240 PAUSE 200: IF INKEY$="s" TH
EN GO TO 99
3250 GO TO 3000
```

FLASH

ZX81

```
   1 REM 12345678901234567890123
4567890
   2 FAST
   5 LET A$="0E0106162A0C40237EF
E762805EE807718F30520F20D20EFC9"
  10 IF LEN A$=50 THEN GOTO 30
  20 PRINT AT 5,0;"HEX CODE LENG
TH ERROR"
  25 STOP
  30 LET START=16514
  40 FOR A=0 TO 24
  50 LET H=CODE A$(2*A+1)-28
  60 LET L=CODE A$(2*A+2)-28
  70 LET N=16*H+L
  80 POKE START+A,N
  90 NEXT A
 100 DIM A$(5,20)
 110 DIM B$(6,27)
 200 LET A$(1)=" "
 210 LET A$(2)=" "
 220 LET A$(3)=" "
```

```
290 LET A$(4)="[graphics]"
300 LET B$(2)="[graphics]"
310 LET B$(3)="[graphics]"
320 LET B$(4)="[graphics]"
330 LET B$(5)="[graphics]"
340 LET B$(6)="[graphics]"
350 LET B$(7)="[graphics]"
390 SLOW
400 FOR N=1 TO 5
410 PRINT AT N+2,6;A$(N)
420 NEXT N
430 FOR N=1 TO 8
460 PRINT AT N+10,3;B$(N)
470 NEXT N
500 REM PAUSE 15
510 GOSUB 600
520 RAND USR 16514
530 GOSUB 600
540 GOTO 400
600 FOR N=1 TO 30
610 NEXT N
620 RETURN
```

POKE LOCATION 16515 FOR NUMBER OF SCREEN FLASHES

POKE LOCATION 16517 FOR NUMBER OF LINES TO REVERSE (1-22)

HEXADECIMAL CODE		ZX MNOMICS	
0E 01		LD C,1	Reverse screen once
06 16	Start:	LD B,22	22 lines
2A 0C 40		LD HL,(D-FILE)	start of display file
23	Top:	INC HL	next character
7E		LD, A,HL	get it
FE 76		CP 118	is it an ENTER?
28 05		JR Z,LINE	if so, skip ahead to LINE
EE 80		XOR 128	reverse character
77		LD (HL),A	replace character
18 F5		JR TOP	do it again
05	Line:	DEC B	next line
20 F2		JR NZ,TOP	if less than 22 lines jump
0D		DEC C	finished reversing?
20 EA		JR NZ,START	if not, jump
C9		RET	otherwise return to basic

2068

POSTER MAKER

Following is a short program to help you make large copies of those fantastic title pages on some of the newer software or, for that mater, your own masterpieces....produced with Top Drawer no doubt. The print out is in the form of four stips wich are then fastened together to produce a poster aproximately 12 x 14 inches. The program, in basic is awfully slow, but it's also easier to understand than some of the machine code programs that do the same thing.

Give it a try

Jim Turner

```
10 CLS : PRINT AT 17,0;"BECAUSE SOME TITLE PAGES USE   DARK PAPER AND LIGHT INK, YOU   WILL HAVE TO DECIDE ON:"
20 PRINT "Positive or Negative"
30 IF INKEY$="n" THEN LET a=1: LET b=0: GO TO 60
40 IF INKEY$="p" THEN LET a=0: LET b=1: GO TO 60
50 GO TO 30
60 CLS : LOAD ""CODE 16384
70 FOR w=0 TO 255 STEP 64
80 FOR s=175 TO 1 STEP -22
90 FOR x=s TO s-21 STEP -2
100 FOR y=w TO w+63 STEP 2
110 IF POINT (y,x)=a AND POINT (y+1,x)=a AND POINT (y,x-1)=a AND POINT (y+1,x-1)=a THEN LPRINT " ";: GO TO 270
120 IF POINT (y,x)=b AND POINT (y+1,x)=b AND POINT (y,x-1)=b AND POINT (y+1,x-1)=b THEN LPRINT "█";: GO TO 270
130 IF POINT (y,x)=b AND POINT (y+1,x)=a AND POINT (y,x-1)=a AND POINT (y+1,x-1)=a THEN LPRINT "▘";: GO TO 270
```

```
 140 IF POINT (y,x)=a AND POINT
(y+1,x)=b AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=a THEN LPRINT
"▝";: GO TO 270
 150 IF POINT (y,x)=a AND POINT
(y+1,x)=a AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=a THEN LPRINT
"▗";: GO TO 270
 160 IF POINT (y,x)=a AND POINT
(y+1,x)=a AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=b THEN LPRINT
"▐";: GO TO 270
 170 IF POINT (y,x)=a AND POINT
(y+1,x)=b AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=b THEN LPRINT
"▟";: GO TO 270
 180 IF POINT (y,x)=b AND POINT
(y+1,x)=b AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=b THEN LPRINT
"▜";: GO TO 270
 190 IF POINT (y,x)=b AND POINT
(y+1,x)=b AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=a THEN LPRINT
"▛";: GO TO 270
 200 IF POINT (y,x)=b AND POINT
(y+1,x)=a AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=b THEN LPRINT
"▙";: GO TO 270
 210 IF POINT (y,x)=a AND POINT
(y+1,x)=b AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=a THEN LPRINT
"▞";: GO TO 270
 220 IF POINT (y,x)=b AND POINT
(y+1,x)=a AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=b THEN LPRINT
"▚";: GO TO 270
 230 IF POINT (y,x)=a AND POINT
(y+1,x)=a AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=b THEN LPRINT
"▄";: GO TO 270
 240 IF POINT (y,x)=b AND POINT
(y+1,x)=b AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=a THEN LPRINT
"▀";: GO TO 270
 250 IF POINT (y,x)=b AND POINT
(y+1,x)=a AND POINT (y,x-1)=b AN
D POINT (y+1,x-1)=a THEN LPRINT
"▌";: GO TO 270
 260 IF POINT (y,x)=a AND POINT
(y+1,x)=b AND POINT (y,x-1)=a AN
D POINT (y+1,x-1)=b THEN LPRINT
"▐";: GO TO 270
 270 NEXT y
 280 NEXT x
 290 NEXT s
 300 LPRINT : LPRINT
 310 NEXT w
```

RELOCATOR - ZX81

Marcel G Boult

If you have written a machine code program that fits above RAMTOP and you now wish to relocate it in a REM statement, then RELOCATOR may be for you.

RELOCATOR will transfer code from one location to another, adjusting CALL's and JP's addresses wherever required. Note that CALL's and JP's that were outside of the original code will not be altered once the code is relocated. Thus, calls or jumps to the ROM will remain intact.

RELOCATOR will handle data or tables within code provided that three additional NOP "guard bytes" are included immediately after the data or tables. There is no guarantee that the data or tables will not be altered; however, the presence of the three "guard bytes" will insure that the code following these will maintain its integrity following relocation.

RELOCATOR may be used to relocate code as well as data. the only difference is that in one case there is alteration while in the other case there is only a block move.

Enter the following HEX loader program thru which the RELOCATOR program code can be entered. Note that the "1 REM" statement must be 450 bytes long.

RELOCATOR uses the zx printer buffer to interface with the user.

Poke 16446 with 40 for code relocation, or with 41 for data transfer.

Poke 16447 and 16448 with the current address of what is to be transferred.

Poke 16449 and 16450 with the new address.

Poke 16453 and 16454 with the size in bytes of what is to be transferred.

Enter "RAND USR 16514" to activate RELOCATOR.

```
  20 REM ** RELOCATOR **
  22 REM ** MAY 11 85
9000 FAST
9001 REM ------------------------
9002 REM HEX LOADER
9003 REM ------------------------
9010 FOR I=16514 TO 16966
9020 INPUT A$
9030 LET HI=(CODE A$(1)-28)*16
9040 LET LO=CODE A$(2)-28
9050 POKE I,HI+LO
9100 NEXT I
9061 STOP
9062 REM ------------------------
9063 REM HEX DISPLAY
9064 REM ------------------------
9066 PRINT "LIST ID?"
9067 INPUT A$
9068 PRINT A$
9070 PRINT "START ADDR "
9075 INPUT A
9076 PRINT A
9077 PRINT "LENGTH "
9078 INPUT B
9079 PRINT B
9080 CLS
9081 FAST
9085 FOR I=A TO A+B
9090 LET P=PEEK I
9095 LET HI=INT (P/16)
9100 LET LO=(P-HI*16)
9105 LET H$=CHR$ (HI+28)
9110 LET L$=CHR$ (LO+28)
9115 PRINT H$;L$;" "
9120 NEXT I
9135 STOP
9190 REM ------------------------
9191 REM RELOCATOR INTERFACE
9192 REM EXAMPLE (RELOCATES RELO
          CATOR ITSELF)
9193 REM START ADDR: 16514
9194 REM DEST ADDR: 28000
9195 REM BLOCK LENGTH: 452
9196 REM TYPE: CODE
9197 REM RELOCATOR ENTRY: 16514
9199 REM ------------------------
9200 POKE 16448,INT (16514/256)
9210 POKE 16447,16514-256*INT (1
6514/256)
9220 POKE 16450,INT (28000/256)
9230 POKE 16449,28000-256*INT (2
8000/256)
9240 POKE 16454,INT (452/256)
9250 POKE 16453,452-256*INT (452
/256)
9255 POKE 16446,40
9260 RAND USR 16514
9270 PAUSE 100
```

```
RELOCATOR 11 MAY 85
START ADDR 16514
LENGTH 452
ED  4B  3F  40  C5  ED  4B  41
40  C5  ED  4B  45  40  C5  3A
3F  40  09  2B  23  43  40  3A
3E  40  D1  E1  C1  C5  E5  D5
AF  ED  42  22  3C  40  CA  40
42  C1  D1  E1  E5  D5  C5  D2
B8  40  ED  B0  18  03  03  2B
EB  09  2B  EB  ED  B8  3A  3E
40  FE  28  C2  40  42  C1  E1
E5  C5  03  7E  FE  DD  CA  D7
40  FE  FD  20  48  20  03  7E
FE  CB  CA  2C  42  FE  21  CA
2C  42  FE  22  CA  F2  41  FE
2A  CA  F2  41  FE  36  CA  2C
42  FE  09  28  25  FE  19  28
21  FE  29  28  1D  FE  33  28
19  FE  23  28  15  FE  2B  28
11  FE  E1  28  0D  FE  E3  28
09  FE  E5  28  05  FE  F9  C2
DE  41  C3  CB  41  FE  ED  28
08  FE  CB  CA  DE  41  C3  4A
41  23  0B  7E  FE  4B  28  15
FE  5B  28  11  FE  7B  28  0D
FE  43  28  09  FE  53  28  05
FE  73  C2  CB  41  C3  F2  41
FE  3F  DA  57  41  FE  D1  D2
57  41  C3  CB  41  FE  00  28
14  FE  02  28  10  FE  12  28
0C  FE  08  28  08  FE  0A  28
04  FE  1A  20  03  C3  CB  41
E6  0F  FE  02  CA  F2  41  FE
0A  CA  F2  41  FE  06  CA  DE
41  FE  0E  CA  DE  41  7E  FE
3F  D2  9F  41  E6  0F  FE  00
CA  DE  41  FE  01  CA  2C  42
FE  08  CA  DE  41  7E  FE  C3
CA  F2  41  FE  D3  CA  DE  41
FE  DB  CA  DE  41  FE  5F  D2
B7  41  C3  CB  41  E6  0F  FE
04  28  0B  FE  0C  28  07  FE
0D  28  03  C3  CB  41  C3  F2
41  23  E5  C5  E1  01  01  00
AF  ED  42  E5  C1  E1  D2  CD
40  C3  40  42  23  23  E5  C5
E1  01  02  00  AF  ED  42  E5
C1  E1  F9  CD  40  C3  40  42
23  E5  5E  23  56  2A  [illegible]  40
AF  ED  52  D2  17  40  [illegible]  40
40  AF  ED  52  DA  17  42  E5
5D  5B  30  40  19  ED  E1  73
23  72  23  10  00  E1  [illegible]  23
E5  C5  E1  01  03  00  AF  ED
42  E5  C1  E1  D2  [illegible]  40  C3
40  42  23  23  23  10  [illegible]  2A
1E  80  23  22  16  40  7E  DD
D9  14  CD  A7  0E  C9  E1  E1
E1  C9  1B  1B  1B
```

The John Oliger Co.
11601 Whidbey Dr.
Cumberland, IN 46229

PRODUCT LIST. All prices include First Class postage/packing-USA and Canada only. Indiana residents add 5% sales tax. Canadian orders must include money order payeable in US $$.

All projects are designed for the ZX computer electronics experimenter/hobbiest with at least some experience at electronics kit building. All pc boards except the power supplies are double sided with edge connector "fingers", and are meant to be used in either expansion board below. All boards are hand made in-house, are G-10 or FR-4 epoxy material + have solder plated traces. All are completely cut and drilled, but none feature plated through holes-feedthrough wires must be installed by the user. There is a 10 day money back satisfaction guarantee on all my products if the product is returned AS RECIEVED in that amount of time. Exchanges will also be honored in the 10 days, too, if sufficient funds for the return postage of the exchanged product are sent with the return. No returns or exchanges will be accepted after 10 days of your orders reciept.

64K DYNAMIC RAM — Bare pc board:$19.95 — Board w/parts ex. ram:$34.95
This project, featured in SQ#3, (Xerox copy avail. @$2.50pp) is a full feature 64K ram. Special features of this memory not found on other 64K memories include: 8-12K on/off switch, 12-16K on/off switch, two banks of 8K ram avail. mapped 8-16K (Bank A/B switch), 48-64K on/off switch. Circuit uses "pin 1 refresh" type 4164 chips for reliable refresh and IS compatable w/TS1500.

PARALLEL PRINTER PORT — Bare pc board:$10.00 — Board w/parts:$20.95
This project was detailed completly in SQ#s 1&2, and the reading of this two part article is a must for the project builder. (Xerox copy of this article avail. @ $5.00pp) The actual port is memory mapped at location FFFFh, and is a bi-directional port. (Both sending to the printer and reading the printer status supported) This board includes only its' schematic, a theory of operation sheet, and a mc listing w/comments of a copy screen routine, for use with the Okidata printers. The circuit and software can be adapted to most other "Centronic" type printers. To achieve full use of the Basic printer commands, part of the Sinclair rom must be patched by putting it on eprom. (Detailed in the article) Thus access to an eprom programmer and reader is a must to make the port user transparent. (Requires rom + ram decode via use of 64K mem, etc)

TMS9918A VIDEO PROJECT/UPGRADE — Bare board "A":$14.95 — Board "A" w/parts:$37.90
Bare board "B":11.95 — Board "B" w/parts:$17.90
Complete package of both boards "A" and "B", along with parts kits for both:$48.95 (Save $6.85!) (Parts kits do NOT incude the TMS9918A VDP, its' 10.7386mhz crystal, or the required 2764 eprom) (These are avail, NOT from myself, at about a total cost of $41.95 to $51.95. Vendor list inc) This project is a very involved upgrade for your TS1000/ZX-81 computer, that allows your ZX to have many features it has never had before. A few quick examples are a SLOW mode almost as fast as fast mode, a BLINK FREE FAST mode, lower case characters option instead of inverse, etc. This project requires a composite type video monitor, the 2764 eprom programmer, and the +5,+12,-5VDC power supply, below. (If you do not already have one) For more details on this one,please write. This project includes COMPLETE documentation, and source listing of the new software/firmware.

TMS9918A VDP ENHANCED BASIC PROGRAMMING CASSETTE for use with the video project:$6.95

2764 EPROM PROGRAMMER — Bare pc board:$10.00 — Board w/parts (ex eprom):$22.95
This programmer is identical to the 2716 programmer above, except for the following: The circuit has been changed to accept 28 pin 2764 eproms ; The eprom is mapped 2000-3FFFh ; 21VDC is now required for the programming voltage. (Avail. from Vpp supply, below) This board comes with its' schematic, assy instructions, and a theory of operation sheet. If this board is ordered along with a 2764 + 2 read board (below), a copy of an unpublished article on using these boards as "cartridge" boards is included with the order at N.C. This programmer can be used on the TS2068

Vpp POWER SUPPLY — Bare pc board:$4.49 — Board w/pc mounting parts:$9.95
This small single sided board provides 4.4, 21, and 25VDC for use with either programmer above. This circuit is designed to put as little stress on the eprom as possible during programming and is recommended for use with the above programmers. Board is supplied with schematic, assembly instructions, and use instructions. Parts kit does NOT include transformer or power cords.

2764 + 2 EPROM READER — Bare board:$10.00 or 3@$25.95 — Board w/parts:$14.95 or 3@$38.95
While small in size and complexity, this board is versatile in that it is capeable of mapping either of its' 2764 eproms anywhere in the 64K map in 8K blocks. (Within reason) This board is intended to be used as a "cartridge", if desired, for storing Sinclair programs on eprom and its' purchase includes a copy of this article if the 2764 programmer is also ordered. This one comes with its' schematic, assy instructions, and a theory of operation sheet. Rom decoding and ram decoding is required. This circuit can itself decode the rom if one jumper is installed.

MEMORY ACCESS MONITOR — Bare pc board:$5.00
This very small board is a very simple circuit that simply indicates via the use of eight LEDs where in the 64K memory map the Z80 cpu is READind or WRITing. The circuit uses only one ic, and is extremely easy to assemble. Although this circuit does not actually do anything, it can be very useful in debugging hardware projects and mc programs that cause the computer to crash, by indicating where the cpu is after the crash. Only a schematic is included with this one because it is simple enough to require only this for construction. This circuit works also on the TS2068

4-SLOT EXPANSION BOARD — Bare pc board:$10.00 — Board w/all conn/sockets:$32.95

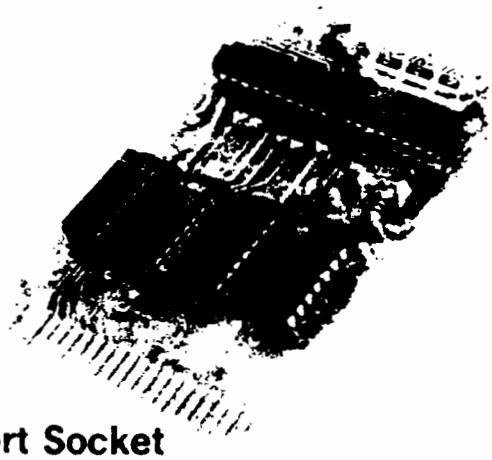
6-SLOT EXPANSION BOARD — Bare pc board:$14.95 — Board w/all conn/sockets:$43.95

Either of these two expansion boards are required to "plug" the other boards above into your system. The 4-slot board features four 25 pin double read out "slots", along with three 20 pin experimental ic socket pads areas. The 6-slot board is almost identical to the 4-slot board, but has six slots and six 20 pin ic socket areas. The complete kits contain all edge connectors and ic sockets that mount on the board, along with one more 23 pin double readout connector for connection to the computer. All edge connectors are pre-cut to size, and have the required slot guide installed. All edge connectors supplied with these boards are prime Texas Instr connectors

Prices subject to change without notice

SOUND GENERATOR for the ZX 81

by Dave Goodman



- ★ **3 Programmable Tone Generators**
- ★ **Noise Generator with 3 Pitch Levels**
- ★ **Separate Attenuators for Noise and Tone Generators**
- ★ **Entry from PEEK and POKE in BASIC**
- ★ **Connects Directly into the Expansion Port Socket (or into the motherboard)**
- ★ **Single Address Access**

This sound generator is a worthy addition to our ZX81 hardware projects. Almost infinite possibilities for sound and noise effects that can be added to your own programs for greater realism.

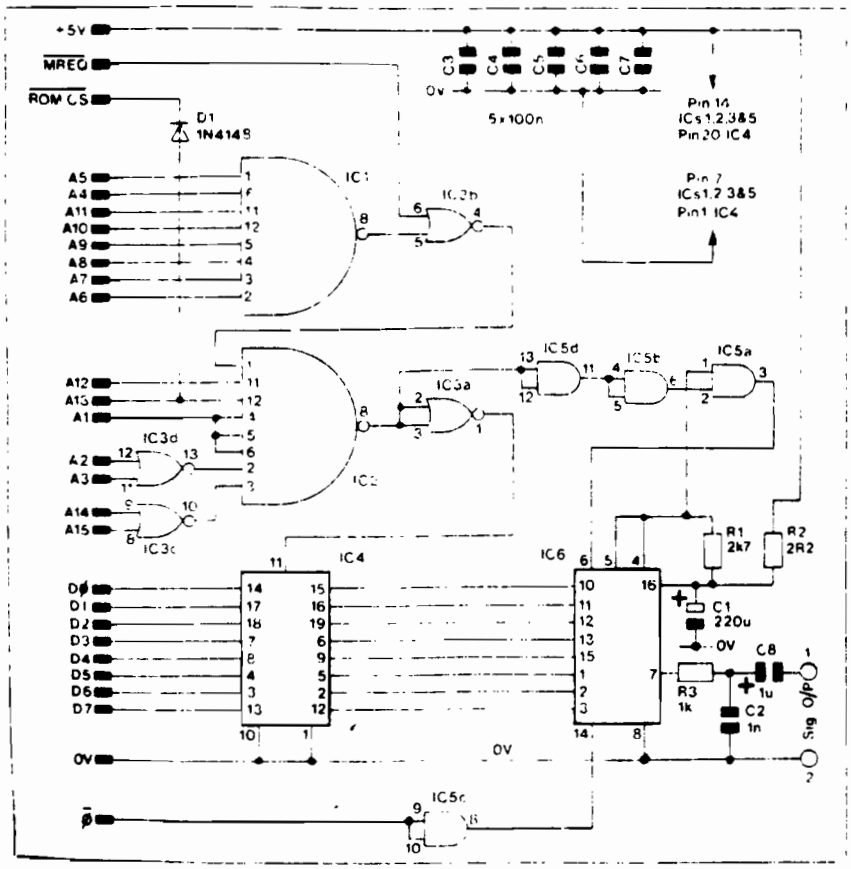
Circuit Description

ICs 1, 2, and 3 are connected to the computer address lines A1 to A15. This means that all addresses up to 65534 may be presented, so a decoder is required to examine all lines, but only to respond to a particular address. The address code used here is 16370, which lies between the 16360 and 16380 used in our I/O port project. A0 is not used, so a further address of 16371 exists.

A negative going address decode pulse appears at the output of IC2, and is used to latch data into IC4 and enable IC6. To avoid corruption of data into IC6 the output of IC4 must be latched to the data code before IC6 is enabled.

Buffers IC5a and b delay the enable pulse just enough to allow IC4 to latch before enabling IC6. IC6 pin 4 READY line controls the duration of the WRITE ENABLE (pin 5) and CHIP ENABLE pulse for correct circuit operation via IC5c. R1 and C1 smoothe the +5V supply to keep noise spikes down to a minimum, and audio output is taken from IC6 pin 7, via low pass filter R3 and C2, to the output pins 1 and 2. IC5d buffers the 3.22MHz clock, and prevents lengthy track runs from crashing the ZX81.

Most important is D1. You may be aware that because of incomplete address decoding the ZX81 ROM is repeated between address 8193 and 16383, which is an unused area between ROM and system variables. These addresses can be POKEd providing that the ROM is deselected at that time, and D1 conducts when A13 is high, freeing this area for use.



Assembly and Construction

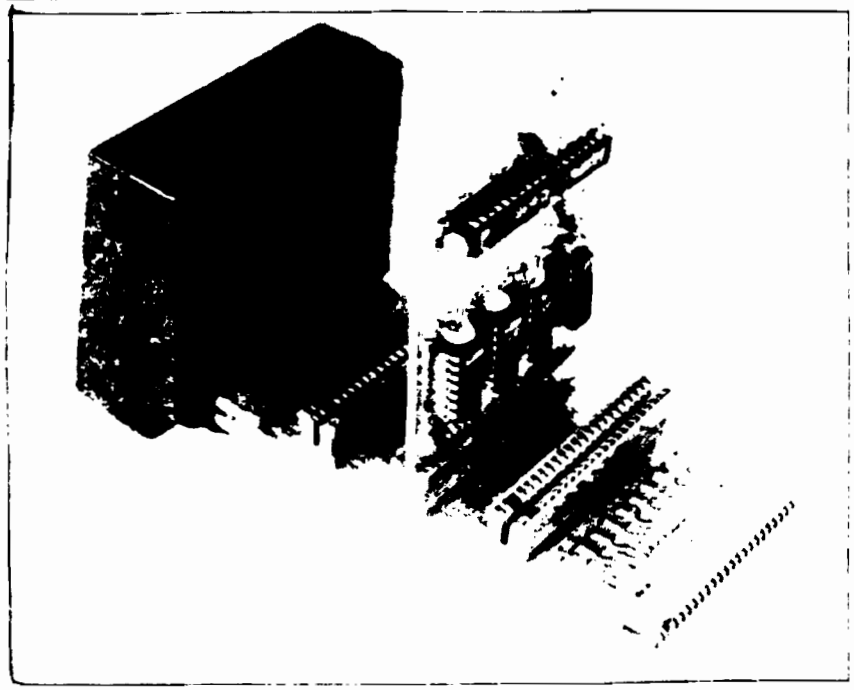
Insert all track pins and both veroboard pins. Fit all six DIL sockets, R1, 2, and 3 and C2. Fit disc ceramics C3 to C7 and C1 and C8 noting the polarity markings. Insert all ICs the correct way round, then clean the PCB and make the final inspection for short circuits and dry joints.

Testing and Use

If you do not possess a mother board you will require a 2 × 23 way socket to solder onto the PCB edge connector, otherwise plug into your motherboard. The signal is insufficient to drive a loudspeaker direct, as it is only 300mV in amplitude, so you will need an external amplifier and speaker connected to pins 1 (signal) and 2 (screen). Switch on the ZX81 and a cacophony of noise should be heard. Run the following test program –

```
10 REM TEST PROGRAM
15 LET A = 16370
20 INPUT B
25 POKE A, B
30 GOTO 10
```

Press RUN then NEWLINE and input the following codes followed by NEWLINE after each code: 159 191 222 255 (you should hear the tones disappearing one by one until all the signals are off) 144 128 64. A low frequency tone of approximately 98Hz (G2)



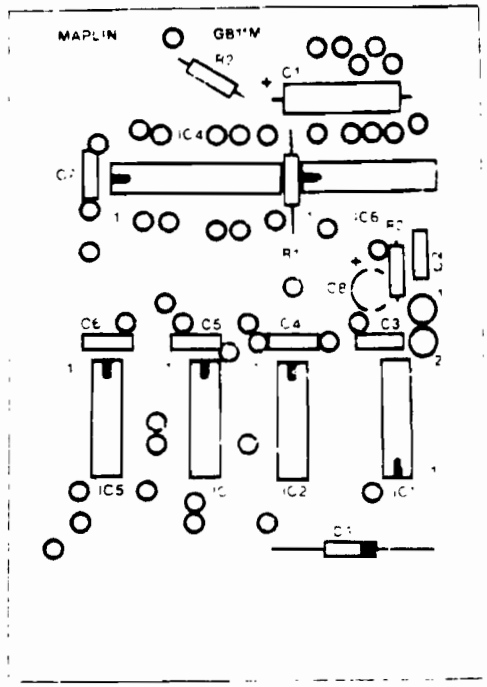


Figure 2 Overlay and artwork

TONE GENERATOR	ACCESS CODE	FREQUENCY RANGE	ATTENUATION CODE
1	128	64(G2) to 1(G8)	144 to 159 (OFF)
2	160	64(G2) to 1(G8)	176 to 191 (OFF)
3	192	64(G2) to 1(G8)	208 to 223 (OFF)
White Noise	228	High pitch	240 to 255 (OFF)
White Noise	229	Med. pitch	240 to 255 (OFF)
White Noise	230	Low pitch	240 to 255 (OFF)
Pulse (spike)	224	480Hz	240 to 255 (OFF)
Pulse (spike)	225	240Hz	240 to 255 (OFF)
Pulse (spike)	226	120Hz	240 to 255 (OFF)

should be heard. Any number between 1 and 64 may now be entered and the appropriate tone should be audible. Refer to the following listing for access codes and settings.

To input data into tone generator 1 only, first enter the required volume level from the attenuation codes. The first code given is maximum volume, e.g. Tone generator 1 = 144 and attenuation levels are in 15 x 2dB steps down to 159 which is fully off. Next enter the tone generator access code, which in this example is 128, followed by the required frequency. The frequency range covers 98Hz (G2) up to 6.3kHz (G8) in 64 steps, code 1 being the highest frequency and code 64 being the lowest, so enter 64. The entered codes are now 144, 128, 64, which is tone generator 1 producing an output of 98Hz at full volume. As a Tone Generator access code was entered after the attenuator, the frequency can be changed as desired, but if an attenuator code is now entered, 144 to 159 in this example, the frequency can only be altered by entering the access code 128 again and then a frequency code.

Keep in mind that when a register is accessed it will remain 'on line' awaiting further update input codes. Access to Tone Generators 2 and 3 is in the same manner except that the codes are different. If you enter 144, 128, and 64 to set up a tone in Generator 1, then enter 176, 160 and 32. Two tones will now be heard, with Generator 2 an octave above Generator 1. Entering a frequency code will now only alter Generator 2.

Pulse and noise effects use the attenuation codes and access codes only. Once the attenuation level has been set, the noise and pulse codes are enabled and are limited only by duty cycle.

Tone Generator 3 can be used to control either noise or pulse registers and code 231 followed by 192 allows control of white noise pitch by entering 1 to 64. Similarly code 227 followed by 192 allows control of a 480Hz pulse tone by entering 1 to 64 and the lowest frequency possible is 5Hz.

Obviously the best way to understand the system is to use it, therefore a few simple programs are given for assistance, shown at the end of this article. When writing music programs remember that G2 to G8 spans 73 notes and control only covers 64 notes, therefore higher frequency notes tend to become sharper in relation to the lower octaves.

Run these programs in SLOW mode —

```
10 REM ZX0878DM
   MDB78DM
   8ZMDB78DM
   EDI NZMDB78
15 LET A = 144
20 LET B = 16514
25 FOR I = 1 TO 4
30 FOR I = 1 TO 4
35 NEXT
40 POKE A, PEEK P
45 LET B = B+1
50 NEXT
55 GOTO 5
Press Break to
stop sequence
10 REM GUN SHOT
15 LET A = 16514
20 POKE A, 229
25 POKE A, 241
30 POKE A, 224
35 POKE A, 24
40 POKE A, 23
45 FOR I = 1 TO 4
50 POKE A
55 NEXT
60 INPUT EE
65 GOTO 10
After RUN press
NEW line
```

```
10 REM PHOTON BLAST
15 LET A = 16514
20 POKE A, 159
25 POKE A, 141
30 POKE A, 223
35 POKE A, 148
40 POKE A, 241
45 POKE A, 231
50 POKE A, 230
55 FOR I = 1 TO 30 STEP 4
60 POKE A
65 POKE A, 128
70 POKE A
75 POKE A
80 POKE A, 192
85 NEXT
90 POKE A, 255
95 POKE A, 15
100 POKE A, 15
105 POKE A, 15
110 GOTO 5
Press NEW LINE to give
To remove a hack
Run 10 plus Photon on
40 and 50 go to
90 to POKE A, 15
```

ZX81 SOUND GENERATOR PARTS LIST

Resistors: All 0.4W 1% Metal Film

R1	2k7		(M2K7)
R2	2R2		(M2R2)
R3	1k		(M1K)

Capacitors

C1	220uF 10V axial electrolytic		(FB60D)
C2	1nF ceramic		(WX62[illegible])
C3-7 inc	100nF minidisc		(YR75S)
C8	1uF35V tantalum	5 off	(WW60Q)

Semiconductors

D1	1N4148		(QL80B)
IC1,2	74LS30		(YF20W)
IC3	74LS02	2 off	(YF02C)
IC4	74LS273		(YH15R)
IC5	74LS08		(YF06G)
IC6	76489AN		(YH33L)

Miscellaneous

14 pin dil skt	4 off	(BL18U)
16 pin dil skt		(BL19V)
20 pin dil skt		(HQ77J)
Veropin 2141		(FL21X)
Track pin	1 pkt	(FL82D)
ZX81 Sound Gen PCB		(GB11M)

A complete kit of parts is available for this project.
Order As LW96E (Sound Gen Kit). Price £10.95

For constructors who may wish to use the Sound Generator in addition to other external hardware, a mother board is available called the ZX81 Extender board (GB08L) and will accept the Sinclair 16k RAM pack and up to 3 plug in modules. In addition to the PCB you will require 4 PC Edge Connectors (RK35C)